

**Original citation:**

Dumont, Thierry and Le Corff, Sylvain. (2014) Simultaneous localization and mapping in wireless sensor networks. Signal Processing: Image Communication, Volume 101 (Number 2). pp. 192-203.

**Permanent WRAP url:**

<http://wrap.warwick.ac.uk/59645>

**Copyright and reuse:**

The Warwick Research Archive Portal (WRAP) makes this work of researchers of the University of Warwick available open access under the following conditions.

This article is made available under the Creative Commons Attribution- 3.0 Unported (CC BY 3.0) license and may be reused according to the conditions of the license. For more details see <http://creativecommons.org/licenses/by/3.0/>

**A note on versions:**

The version presented in WRAP is the published version, or, version of record, and may be cited as it appears here.

For more information, please contact the WRAP Team at: [publications@warwick.ac.uk](mailto:publications@warwick.ac.uk)



# Simultaneous localization and mapping in wireless sensor networks



Thierry Dumont<sup>a,1</sup>, Sylvain Le Corff<sup>b,\*,2</sup>

<sup>a</sup> Laboratoire MODALX, Université Paris Ouest Nanterre La Défense, France

<sup>b</sup> Department of Statistics, University of Warwick, United Kingdom

## ARTICLE INFO

### Article history:

Received 28 February 2013

Received in revised form

24 January 2014

Accepted 14 February 2014

Available online 22 February 2014

### Keywords:

Simultaneous localization and mapping

Indoor localization

Received signal strength indicator

Parameter estimation

Signal propagation

## ABSTRACT

Mobile device localization in wireless sensor networks is a challenging task. It has already been addressed when the WiFi propagation maps of the access points are modeled deterministically or estimated using an offline human training calibration. However, these techniques do not take into account the environmental dynamics. In this paper, the maps are assumed to be made of an average indoor propagation model combined with a perturbation field which represents the influence of the environment. This perturbation field is embedded with a distribution describing the prior knowledge about the environmental influence. The device is localized with Sequential Monte Carlo methods and relies on the estimation of the propagation maps. This inference task is performed online, using the observations sequentially, with a new online Expectation Maximization based algorithm. The performance of the algorithm is illustrated with Monte Carlo experiments using both simulated data and a true data set.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/3.0/>).

## 1. Introduction

In this paper, we consider a WiFi communication network made up of a mobile device, a server and WiFi access points (AP). In this context, a key step to localize the mobile device is to estimate the WiFi signal strength at different positions in the environment. However, in an indoor environment, signals may experience complex attenuation such as shadowing or reflection.

Different techniques can be used to approximate the WiFi propagation map of each AP. In Gorce et al. [19], a deterministic model based on the characteristics of the surrounding AP and on the localization of the obstacles in the

environment is introduced. In Bahl and Padmanabhan [2] and Evvinnou and Marx [16], a previous offline training phase is performed. In this site survey, the signal strength indicator (RSSI) received from different AP is measured at some previously determined positions. This allows us to build an accurate estimation of the signal strength, but only for a finite number of points. Ferris et al. [17] provide a method to extend these measures to the entire map using Gaussian processes techniques.

In this paper, we propose an estimation method that does not require any calibration procedure. The propagation maps are estimated online, without storing the observations, using the data sent by the mobile device. Any modification in the signal propagation (due to new obstacles for instance) affects the data sent by the mobile device. Then, while these changes deteriorate the accuracy of localization systems based on fixed estimators of the propagation maps, our estimation procedure takes them into account. Thus, as illustrated in Section 5.2, the accuracy of our localization method improves with time instead of degrading.

\* Corresponding author.

E-mail addresses: [thierry.dumont@u-paris10.fr](mailto:thierry.dumont@u-paris10.fr) (T. Dumont), [s.y.le-corff@warwick.ac.uk](mailto:s.y.le-corff@warwick.ac.uk) (S. Le Corff).

<sup>1</sup> This work is partially supported by ID Services, 91400 Orsay, France.

<sup>2</sup> The author would like to acknowledge research funding from EPSRC through the Centre for Research in Statistical Methodology.

We use a semiparametric statistical model introduced in [15, Chapter 5]: the propagation maps are made of a parametric average indoor model and a nonparametric perturbation field. This model combines a prior knowledge on signal propagation with random perturbations due to the obstacles. Based on the data collected by the mobile device, the parameters and the perturbation field can be estimated. The proposed procedure relies on a new online Expectation Maximization (EM) based algorithm introduced in Le Corff and Fort [21,22] and on Sequential Monte Carlo methods. The device position can be simultaneously estimated using the weighted samples produced by the Sequential Monte Carlo method.

The structure of this paper is the following. Section 2 details the approach of the paper in comparison with other methods for localization and mapping in wireless sensor networks. Section 3 describes the model and defines the notations. Section 4 introduces our algorithm for the considered Simultaneous Localization and Mapping (SLAM) problem. Section 5 illustrates the performance of this algorithm with numerical experiments.

## 2. State of the art and main contributions

### 2.1. Wireless sensor networks

In a radio network field, the signal strength is measured by the RSS (*received signal strength*, in dBm). Each WiFi connected device can compute the RSS as it is needed to associate the device with the AP providing the best signal/noise ratio. Localization systems based on RSS measurements allow us to locate any WiFi connected device.

**Remark 1.** Different devices might have different methods to compute the RSS. It is common to use the terminology of RSSI (*received signal strength indicator*) expressed without unit to name the information on the signal level provided by a WiFi device. To overcome the issue of information disparity between WiFi connected devices, a previous RSSI to RSS conversion might be needed for each localized device. The conversion rule is specific to the device's WiFi card. For the sake of clarity, we assume in this paper that the mobile device's RSSI corresponds to the standard RSS.

Using RSS to estimate the position of the device is challenging. As represented in Fig. 1, RSS is highly unstable as it fluctuates considerably between two consecutive measures at the same position. It is common to describe the RSS variations using a Gaussian representation.

Despite the instant variations of the RSS, its mean value strongly depends on the position of the device. The function that returns the expected RSS at each position is called the *propagation map*. In free spaces, signals propagate in straight line from the emitter to the receptor. The propagation maps are isotropic and can be described using few parameters, see Friis [18]. Therefore, there exist two parameters  $c$  and  $d$  such that the strength of a signal received at  $x$  and emitted at  $O$  is  $c + d \log(\|x - O\|)$  where  $c$  and  $d$  depend on the network characteristics.

On the other hand, indoor propagation of WiFi signals is not isotropic. Fig. 2 represents the *propagation map* of a WiFi signal in an indoor environment. This figure was built deterministically by Gorce et al. [19] using the physical properties of electromagnetic signals.

#### 2.1.1. General model for signal propagation

Let  $X_k$  be the device position at time  $k$ . The received signal strength vector measured by the device, denoted by  $Y_k$ , is written as

$$Y_k = F_k^*(X_k) + \epsilon_k. \quad (1)$$

In the following, the superscript  $\star$  is used to name the true value of every unknown parameter involved in the model.  $\{\epsilon_k\}_{k \geq 0}$  are i.i.d. multidimensional random variables with distribution  $\mathcal{N}(0, \sigma^{*2} Id)$  where  $Id$  is the identity matrix. Each component of  $F_k^*(x)$  represents the expected RSS at position  $x$  relative to an AP. The time dependency of  $F^*$  brings the environmental effects on the propagation maps into relief. The distribution of  $\{\epsilon_k\}_{k \geq 0}$  implies that the noise affecting the RSS of different AP are independent. This assumption is somehow strong, however, the correlation between the RSS of different AP can be hardly taken into account and can strongly depend on the environment configuration.

#### 2.1.2. Contribution of the paper and comparison with the state of the art

Indoor localization requires to overcome two challenges. One has to design a good approximation of  $F_k^*$  and then to use this approximation to estimate the position  $X_k$  corresponding to a given measure  $Y_k$ .

The contribution of this paper is a new estimation procedure of the propagation maps. This new method is combined to a localization algorithm to highlight the relevance of our estimation procedure. Using our propagation maps estimator, many positioning algorithms may be considered. We do not address a comparison of the accuracy between our method and the state of art but a new way to improve indoor localization deployment at a large scale.

$F_k^*$  can be approximated deterministically using physical properties (see [19,26]). Such constructions need a precise description of the environment such as the position and the composition of the walls and furniture present in the environment. They also need a fine knowledge about the different factors influencing signal propagation such as the humidity level in the environment. Without this precise description, the propagation maps built using deterministic methods may be far from reality.

Most of the existing WiFi based localization systems use a preliminary measurement campaign (offline) which is also called fingerprinting technique. A human operator performs a site survey by measuring the RSS from different AP at some fixed positions in the environment. This set of measures, associated with positions, can be directly stored in a database [2] to be used in the localization system as a reference. These measures can also be extrapolated to the entire map using kriging methods such as in Ferris et al. [17]. Users must compile a fairly dense radio map comprising many RSS measurements at many sampled points to attain reasonable positioning accuracy. Moreover, these methods suppose that the propagation maps remain constant which, with our

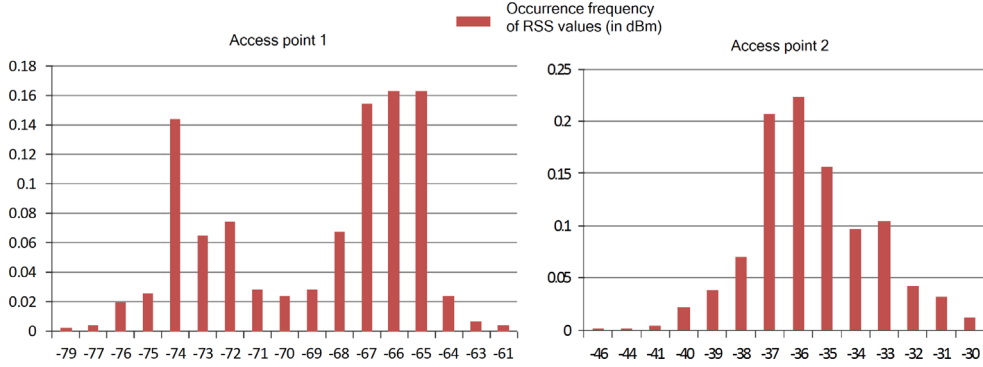


Fig. 1. Histograms of the RSS frequencies for two AP.

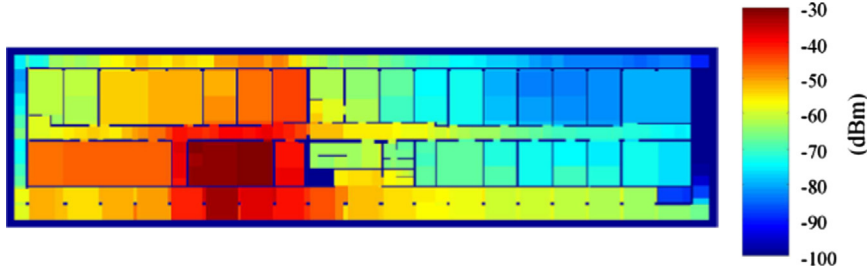


Fig. 2. Representation of a WiFi propagation map deterministically computed.

notation, means that the propagation function  $F^*$  does not depend on time. The measurement campaign can be seen as an instant “photograph” of the propagation map and has to be regularly performed in order to maintain the accuracy of the system. These problems have particularly been spotted by Chen et al. [9] and Madigan et al. [24]. Chen et al. [9] introduce RFID sensors in the environment to perform passive site survey. Madigan et al. [24] use a hierarchical Bayesian model to localize the mobile without site survey but rely on isotropic propagation maps which might lead to a bad estimation in complex indoor environments (although they study a more elaborate model that includes “corridors effects”).

In this paper, we present a new estimation procedure of the propagation maps which does not involve any measurement campaign or additional sensors. The considered model does not assume any knowledge on the environment apart from the position of the AP. The propagation maps are estimated using an online Expectation Maximization (EM) based algorithm using the RSS measurements collected by the device. A similar approach can be found in Chai and Yang [8] which uses the classical EM algorithm to refine the propagation maps estimators obtained using a preliminary site survey. The first substantial benefit of our method concerns its ability to be implemented in any building. Moreover, unlike fingerprinting based methods, the precision of our localization system does not degrade with time as each measure  $Y_k$  is used to improve the propagation maps estimators. The computation of the estimators requires *sufficient statistics*. These statistics summarize the information contained in all the past measures since they were last reset. If there are regular environmental modifications that strongly affect the WiFi propagation, regularly resetting these sufficient statistics is

enough to improve the estimation. Then, our system is more robust than site survey based methods using static propagation maps estimators.

Once the expected RSS has been estimated for the whole map or for some fixed positions, the device can be localized. Bahl and Padmanabhan [2] use the nearest neighbor algorithm. With this algorithm, the strong variability of the RSS leads to a very unstable localization. We use particle filtering to track the mobile device. Such filters have already been used by Ferris et al. [17] in a similar way. Evennou and Marx [16] introduce particle filtering on the Voronoi diagram. Such filters provide a much more stable sequence of positions despite the high variability of the data.

**Remark 2.** There is no chance to identify  $\{F_k^*\}_{k \geq 0}$  with the observations  $\{Y_k\}_{k \geq 0}$  only. In the next section, we omit the time dependency of  $F^*$  which might seem contradictory with our introduction. However, as stated in the above section, regularly resetting the *sufficient statistics* allows us to adapt the algorithm to environmental changes.

### 3. Model and assumptions

Let  $X_k$  be the cartesian coordinates of the mobile device at time  $k$  in a two-dimensional compact space. This compact space represents the map of the one-floor building where the localization is performed. This continuous environment is discretized into a finite grid map, denoted by  $\mathcal{C}$ . It is assumed that  $\{X_k\}_{k \geq 1}$  is a Markov chain taking values in  $\mathcal{C}$  with initial distribution  $\nu$  and Markov transition matrix given, for all  $(x, x') \in \mathcal{C}^2$ , by

$$q(x, x') \propto e^{-\|x - x'\|^2/a}, \quad (2)$$

where  $\|\cdot\|$  denotes the usual euclidean norm in  $\mathbb{R}^2$  (the associated inner product is denoted by  $\langle \cdot, \cdot \rangle$ ).  $a \in \mathbb{R}_+^*$  depends on the average speed of the mobile and is assumed to be known. Therefore, the initial state  $X_0$  is distributed according to  $\nu$  and, for any  $k \geq 1$  and any  $x \in C$ , given  $X_k = x$ ,  $X_{k+1} = x'$  with probability  $q(x, x')$ .

Let  $B$  be the number of AP and  $|C|$  be the cardinality of  $C$ . In the sequel, for any  $B \times \mathbb{R}^{|C|}$  matrix  $A$ , we use the shorthand notation  $A_j$  for the vector  $\{A_{j,x}\}_{x \in C}$  and  $A_j^2$  for the vector  $\{A_{j,x}^2\}_{x \in C}$ . At each time step  $k \geq 1$  and for each  $j \in \{1, \dots, B\}$ , the mobile device measures and sends to the server the observation  $Y_{k,j}$  given by

$$Y_{k,j} \stackrel{\text{def}}{=} \mu_{j,x_k}^* + \delta_{j,x_k}^* + \varepsilon_{k,j}, \quad (3)$$

where

- $\mu_{j,x}^*$  is the  $j$ -th average indoor propagation term at position  $x$ . For all  $x \in C$  and all  $j \in \{1, \dots, B\}$ ,  $\mu_{j,x}^*$  only depends on the distance between  $x$  and  $O_j$  where  $O_j$  is the known position of the  $j$ -th AP. In the sequel, we use the so-called Friis transmission equation given by Friis [18]:

$$\mu_{j,x}^* \stackrel{\text{def}}{=} c_j^* + d_j^* \log \|x - O_j\|, \quad (4)$$

where  $c_j^*$  and  $d_j^*$  are parameters depending on the environment and where  $\log$  is the logarithm to the base  $e$ .

- $\delta_j^*$  is an additive term due to random perturbations. A similar model of WiFi propagation maps using Gaussian processes can be found in Ferris et al. [17]. It is assumed that the parameters  $\{\delta_j^*\}_{j=1}^B$  are embedded with the prior distribution  $\pi$  given, for any  $\delta \in \mathbb{R}^{B|C|}$ , by

$$\pi(\delta) \propto \exp \left\{ -\frac{1}{2} \sum_{j=1}^B \delta_j^T \Sigma^{-1} \delta_j \right\},$$

where for any matrix  $A$ ,  $A^T$  is the transpose of  $A$  and where  $\Sigma$  is assumed to be known.

- $\{\varepsilon_k\}_{k \geq 0}$  is a sequence of i.i.d Gaussian random vectors in  $\mathbb{R}^B$ , independent of  $\{X_k\}_{k \geq 1}$ , with mean 0 and covariance matrix  $\sigma^{*,2} I_B$ , where  $I_B$  is the identity matrix of size  $B \times B$ .

$F_j \stackrel{\text{def}}{=} \mu_j^* + \delta_j^*$  will be referred to as the true propagation map of the  $j$ -th AP and  $F^* \stackrel{\text{def}}{=} \{F_j^*\}_{j=1}^B$  as the true propagation maps. Fig. 3 displays a realization of  $\delta_j^*$  (sampled from  $\pi$ ) and the functions  $\mu_j^*$  and  $F_j^*$ , defined on the grid  $C = \{0, \dots, 30\} \times \{0, \dots, 30\}$ . The parameters used in this figure are  $O_j = (15, 15)$ , and  $c_j^*$ ,  $d_j^*$  and  $\Sigma$  are given in Section 5. In the sequel, we write

$$\theta^* \stackrel{\text{def}}{=} (c^*, d^*, \delta^*, \sigma^{*,2}),$$

where

$$c^* \stackrel{\text{def}}{=} \{c_j^*\}_{j=1}^B, \quad d^* \stackrel{\text{def}}{=} \{d_j^*\}_{j=1}^B \quad \text{and} \quad \delta^* \stackrel{\text{def}}{=} \{\delta_j^*\}_{j=1}^B.$$

For any  $x \in C$  and any  $k \geq 1$ , the distribution of  $Y_k$  conditionally to  $X_k = x$  has a density with respect to the Lebesgue measure on  $\mathbb{R}^B$  given by  $g_{\theta^*}(x, Y_k)$ , where, for

$$\text{all } y \stackrel{\text{def}}{=} (y_1, \dots, y_B) \in \mathbb{R}^B,$$

$$g_{\theta^*}(x, y) \stackrel{\text{def}}{=} (2\pi\sigma^{*,2})^{-B/2} \prod_{j=1}^B \exp \left\{ -\frac{|y_j - F_{j,x}^*|^2}{2\sigma^{*,2}} \right\}.$$

In the sequel, we aim at simultaneously estimating the device position and  $\theta^*$  using the observations  $\{Y_k\}_{k \geq 1}$ . For any positive integer  $n$ , any observation set  $(y_1, \dots, y_n)$  shortly denoted by  $y_{1:n}$ , and any parameter  $\theta = (c, d, \delta, \sigma^2)$ , the likelihood of the observations  $y_{1:n}$  is given by

$$L_{\theta}(y_{1:n}) \stackrel{\text{def}}{=} \sum_{x_{1:n} \in C^n} \nu(x_1) g_{\theta}(x_1, y_1) \prod_{k=2}^n q(x_{k-1}, x_k) g_{\theta}(x_k, y_k).$$

Let  $n$  be a positive integer and  $Y_{1:n}$  be a set of observations. The estimator of  $\theta^*$  is set as one maximizer of the function:

$$\theta = (c, d, \delta, \sigma^2) \mapsto n^{-1} [\log L_{\theta}(Y_{1:n}) + \log \pi(\delta)]. \quad (5)$$

## 4. Online estimation procedure

### 4.1. EM based algorithms to estimate the propagation maps

The EM algorithm is a well-known iterative algorithm to perform maximum likelihood estimation in hidden Markov models [13]. An EM based algorithm can be introduced to maximize (5). Each iteration of this algorithm is decomposed into an E-step where the expectation of the complete data log-likelihood (log of the joint distribution of the states and the observations) conditionally on the observations is computed; and a M-step which updates the parameter estimate. Let  $Y_{1:n}$  be a fixed set of observations and  $\theta_i$  be the current map estimate.

- The E-step amounts to computing the conditional expectation

$$Q_{\theta_i}(Y_{1:n}; \theta) = \mathbb{E}_{\theta_i} \left[ \frac{1}{n} \log p_{\theta}(X_{1:n}, Y_{1:n}) | Y_{1:n} \right], \quad (6)$$

where  $\log p_{\theta}(X_{1:n}, Y_{1:n})$  is the complete data log-likelihood and  $\mathbb{E}_{\theta_i}[\cdot | Y_{1:n}]$  is the conditional expectation given  $Y_{1:n}$  when the map is  $\theta_i$ .

- The M-step computes the new value  $\theta_{i+1}$  by choosing one of the maps  $\theta$  maximizing

$$\theta = (c, d, \delta, \sigma^2) \mapsto Q_{\theta_i}(Y_{1:n}; \theta) + n^{-1} \log \pi(\delta).$$

Define, for any  $(x, y) \in C \times \mathbb{R}^B$  and any  $j \in \{1, \dots, B\}$ , the vectors

$$s_1(x) \stackrel{\text{def}}{=} \{1_{x'}(x)\}_{x' \in C}, \quad s_{2,j}(x, y) \stackrel{\text{def}}{=} \{1_{x'}(x) y_j\}_{x' \in C}, \quad s_{3,j}(y) \stackrel{\text{def}}{=} y_j^2,$$

where  $1_{x'}(x)$  equals 1 if  $x = x'$  and 0 otherwise. The constant  $a$  being known, the intermediate quantity associated with the model presented in Section 3 can be written, up to an additive constant, as

$$Q_{\theta_i}(Y_{1:n}; \theta) = -\frac{B}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{j=1}^B \{ \langle S_1, F_j^2 \rangle - 2 \langle S_{2,j}, F_j \rangle + S_{3,j} \}, \quad (7)$$

where (the dependence on  $\theta_i$ ,  $n$  and  $Y_{1:n}$  is dropped from the notation for better clarity)

$$S_1 \stackrel{\text{def}}{=} \mathbb{E}_{\theta_i} \left[ \frac{1}{n} \sum_{k=1}^n s_1(X_k) | Y_{1:n} \right], \quad S_{2,j} \stackrel{\text{def}}{=} \mathbb{E}_{\theta_i} \left[ \frac{1}{n} \sum_{k=1}^n s_{2,j}(X_k, Y_k) | Y_{1:n} \right],$$

$$S_{3,j} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{k=1}^n s_{3,j}(Y_k).$$

Therefore, by (7), it is enough to compute  $S_1$ ,  $S_{2,j}$  and  $S_{3,j}$ ,  $1 \leq j \leq B$ , to maximize the function  $\theta = (c, d, \delta, \sigma^2) \mapsto Q_{\theta_i}(Y_{1:n}; \theta) + n^{-1} \log \pi(\delta)$ . The detailed computations to solve this optimization problem and to obtain the new map estimate  $\theta_{i+1}$  are given in Algorithm 1 (where  $\mathbb{1}$  is the vector of size  $|\mathcal{C}|$  where each entry equals 1 and, for any vector  $v$ ,  $\text{diag}(v)$  is the diagonal matrix with diagonal given by  $v$ ).

**Algorithm 1.** Map update.

**Require:**  $n$ ,  $S_1$ ,  $\{S_{2,j}\}_{j=1}^B$ ,  $\{S_{3,j}\}_{j=1}^B$ .  
 1: *Computation of intermediate quantities*  
 2: **for**  $j=1$  to  $B$  **do**  
 3:   **for**  $x \in \mathcal{C}$  **do**  
 4:      $D_{j,x} = \log \|x - O_j\|$ .  
 5:   **end for**  
 6:    $M_{0,j} = \text{diag}(S_1) + \frac{\sigma^2}{n+1} \Sigma^{-1}$ .  
 7:    $M_{1,j} = \text{diag}(S_1)[I_{|\mathcal{C}|} - M_{0,j}^{-1} \text{diag}(S_1)]$ .  
 8:    $M_{2,j} = I_{|\mathcal{C}|} - \text{diag}(S_1)M_{0,j}^{-1}$ .  
 9:    $W_{1,j} = \mathbb{1}^T M_{1,j} \mathbb{1}$ .  
 10:    $W_{2,j} = \mathbb{1}^T M_{1,j} D_j$ .  
 11:    $W_{3,j} = D_j^T M_{1,j} D_j$ .  
 12:    $w_j = W_{1,j} W_{4,j} - W_{2,j}^2$ .  
 13: **end for**

14: *Parameter update*

15: **for**  $j=1$  to  $B$  **do**

16:    $c_j = w_j^{-1} [W_{3,j} \mathbb{1}^T - W_{2,j} D_j^T] M_{2,j} S_{2,j}$ .

17:    $d_j = w_j^{-1} [-W_{2,j} \mathbb{1}^T + W_{1,j} D_j^T] M_{2,j} S_{2,j}$ .

18:    $\delta_j = M_{0,j} [S_{2,j} - \text{diag}(S_1)(c_j \mathbb{1} + d_j D_j)]$ .

19:    $F_j = c_j \mathbb{1} + d_j D_j + \delta_j$ .

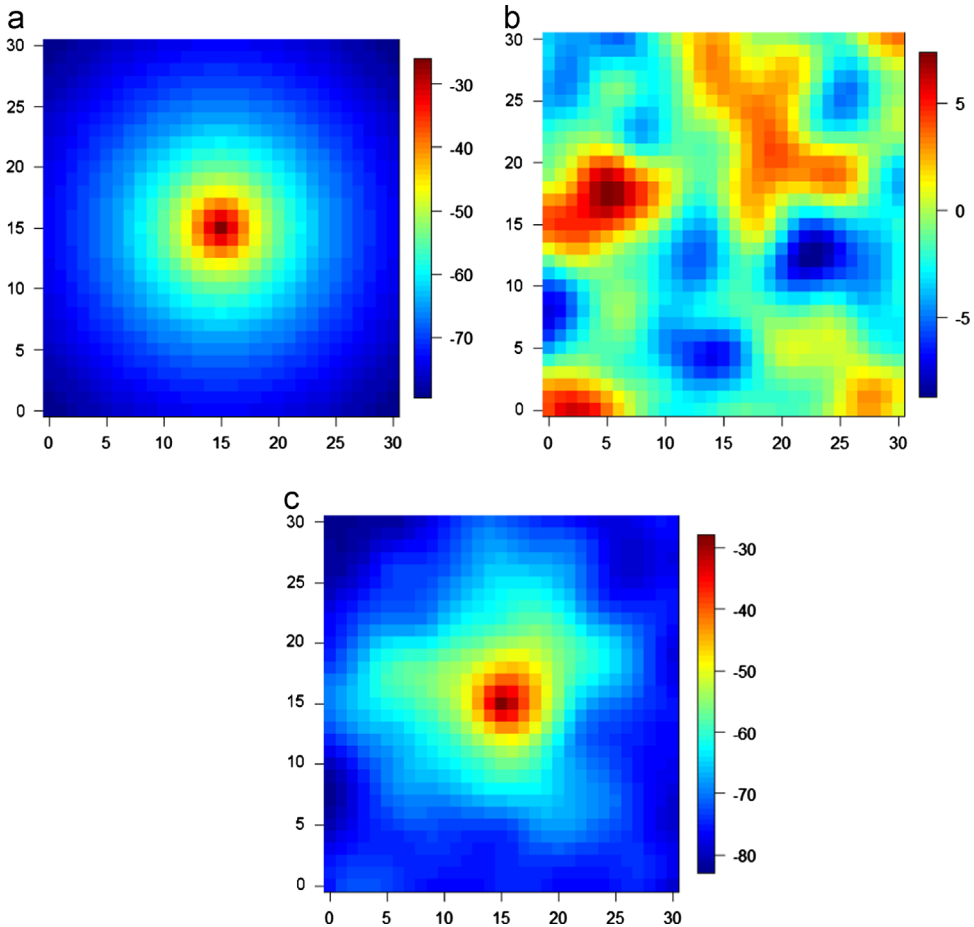
20:    $\sigma^2 = B^{-1} \sum_{j=1}^B \{F_j^T \text{diag}(S_1) F_j - 2S_{2,j}^T F_j + S_{3,j}\}$ .

21: **end for**

22: **return**  $\theta_{i+1} = (c, d, \delta, \sigma^2)$ .

This two step process can be repeated until the likelihood does not improve significantly. However, when the observations are obtained sequentially, this algorithm does not produce a new estimate as new observations are received. The mobile device localization requires an online method which does not store the data and which frequently updates the propagation maps.

Online variants of the EM algorithm have been proposed to obtain map estimates each time a new observation is available. In the case of i.i.d. observations, Cappé and Moulines [6] proposed the first EM based online algorithm. This algorithm replaces the exact computation of the sufficient statistics  $S_1$ ,  $S_{2,j}$  and  $S_{3,j}$  by a stochastic approximation step. In the case of hidden Markov models, when both the



**Fig. 3.** Example of  $\delta_j^*$  (sampled from  $\pi$ ) and of the functions  $\mu_j^*$  and  $F_j^* = \mu_j^* + \delta_j^*$  (in dBm).



observations and the states take a finite number of values (resp. when the state-space is finite) an online EM-based algorithm was proposed by Mongillo and Denève [25] (resp. by [5]). These algorithms combine an online approximation of the filtering distributions of the hidden states and a stochastic approximation step to compute an online approximation of the sufficient statistics. This has been extended to the case of general state-space models with Sequential Monte Carlo algorithms in Cappé [4], Del Moral et al. [12] and Le Corff et al. [23]. More recently, Le Corff and Fort [21,22] proposed the Block Online EM (BOEM) algorithm in which the estimate is kept fixed as block of observations is received and is updated at the end of each block. See also Andrieu et al. [1] for an overview of online estimation procedures using Sequential Monte Carlo methods.

#### 4.2. Proposed algorithm for online localization in wireless sensor networks

This paper introduces an EM algorithm for online localization in wireless sensor networks based on the algorithm introduced in Le Corff and Fort [21,22]. Let  $\{\tau_k\}_{k \geq 1}$  be a sequence of block-sizes and define  $T_0 \stackrel{\text{def}}{=} 0$  and, for any  $k \geq 1$ ,  $T_k \stackrel{\text{def}}{=} \sum_{i=1}^k \tau_i$ . Within each block of observations  $\mathbf{Y}_k \stackrel{\text{def}}{=} Y_{T_{k-1}+1:T_k}$ , the estimate  $\theta_k$  is kept fixed and the sufficient statistics  $S_1^k$ ,  $S_{2,j}^k$  and  $S_{3,j}^k$  are computed sequentially using the current estimate  $\theta_k$ , the observations  $\mathbf{Y}_k$  and  $\tau_{k+1}$  as the number of observations. The superscript  $k$  indicates which observations are used in the definition of the statistics. The next parameter estimate  $\theta_{k+1}$  is computed when the last observation  $Y_{T_{k+1}}$  is received using Algorithm 1.

Unlike in the traditional EM algorithm where the sufficient statistics are computed using forward-backward techniques, Cappé [5], Del Moral et al. [12] and Le Corff and Fort [22] proposed to compute the sufficient statistics recursively (i.e. as the observations are received and without any storage). In general state-space hidden Markov models, these online computations are not available in closed form (except in simple models such as linear Gaussian models) and have to be approximated, e.g. using sequential Monte Carlo methods [4,12]. For finite state-space hidden Markov models, the computations can be performed in theory but are computationally too expensive if the number of states is too large (which is the case in our localization framework, see Section 5). Therefore, sequential Monte Carlo methods are used in this paper to localize the mobile.

In this case, the filtering distribution  $\phi_t^k$  on the block  $k$ , i.e. the distribution of  $X_{T_k+t}$  given  $Y_{T_{k-1}+1:T_k+t}$  and  $X_{T_k} \sim \nu_N$  is approximated by weighted particles  $\{(\xi_{T_k+t}^i, \omega_{T_k+t}^i)\}_{i=1}^N$  such that

$$\hat{\phi}_t^k(x) = \sum_{i=1}^N \omega_{T_k+t}^i \delta_{\xi_{T_k+t}^i}(x),$$

where  $\delta_{\xi_{T_k+t}^i}$  denotes the Dirac distribution at position  $\xi_{T_k+t}^i$ .

For all  $k \geq 0$  and  $t \in \{1, \dots, \tau_{k+1}\}$ ,  $\{\xi_{T_k+t}^i\}_{i=1}^N$  is a set of possible mobile positions at time  $T_k+t$ . At each time step, the new population of particles is built from the previous population using the *bootstrap filter*, see Gordon et al. [20]. The Bootstrap filter combines sequential importance sampling and

resampling steps to produce this set of random particles with associated importance weights. Implementations of such procedures are detailed in Cappé [3], Del Moral [11], Cappé et al. [7], and Doucet and Johansen [14].

*Online map estimation:* We describe here the online approximation on the block  $\mathbf{Y}_k$  of the statistic  $S_1^k$  which is used to compute the map estimate  $\theta_{k+1}$  when  $T_{k+1}$  is received. The computations for  $S_{2,j}^k$  and  $S_{3,j}^k$  follow the same lines. The rationale to establish this online approximation can be found in Del Moral et al. [12]. The first particles  $\xi_0^i$ ,  $i \in \{1, \dots, N\}$ , are sampled uniformly in  $\mathcal{C}$  and the first weights are set as  $\omega_0^i = N^{-1}$ ,  $i \in \{1, \dots, N\}$ :

- (i) Set  $\rho_{T_k}^i = 0$  for all  $i \in \{1, \dots, N\}$ .
- (ii) For all  $t \in \{1, \dots, \tau_{k+1}\}$  repeat

- (a) For all  $i \in \{1, \dots, N\}$ ,

- draw  $l$  in  $\{1, \dots, N\}$  with probabilities  $\{\omega_{T_k+t-1}^{\ell}\}_{\ell=1}^N$ ;
  - sample  $\xi_{T_k+t}^i \sim q(\xi_{T_k+t-1}^l, \cdot)$ ;
  - set  $\omega_{T_k+t}^i \propto g_{\theta_k}(\xi_{T_k+t}^i, Y_{T_k+t})$ .
- Compute

$$(b) \quad \rho_{T_k+t}^i = \frac{1}{t} s_1(\xi_{T_k+t}^i) + \frac{t-1}{t} \sum_{\ell=1}^N \frac{\omega_{T_k+t-1}^{\ell} q(\xi_{T_k+t-1}^{\ell}, \xi_{T_k+t}^i) \rho_{T_k+t-1}^{\ell}}{\sum_{p=1}^N \omega_{T_k+t-1}^p q(\xi_{T_k+t-1}^p, \xi_{T_k+t}^i)}.$$

- (iii) The approximation of  $S_1^k$  on the block  $\mathbf{Y}_k$  is then given by

$$\hat{S}_1^k = \sum_{i=1}^N \omega_{T_k+\tau_{k+1}}^i \rho_{T_k+\tau_{k+1}}^i.$$

- (iv) Once these computations are done for each statistic, the estimate  $\theta_{k+1}^k$  is computed by Algorithm 1 applied with  $S_1^k$ ,  $S_{2,j}^k$ ,  $S_{3,j}^k$  and  $\tau_{k+1}$ .

The BOEM proposed in Le Corff and Fort [22] also introduced an averaged estimate  $\{\bar{\theta}_k\}_{k \geq 0}$  based on a weighted mean of all the sufficient statistics computed in the past. It is proved in Le Corff and Fort [22] that this averaged estimator has an optimal rate of convergence. This can be easily done recursively after the computation of each statistic in step (iii). Step (iii) is then followed by (iii') Compute (with  $\hat{S}_1^0 = 0$ )

$$\hat{S}_1^k = \frac{T_k}{T_{k+1}} \hat{S}_1^{k-1} + \frac{\tau_{k+1}}{T_{k+1}} \hat{S}_1^k.$$

And step (iv) is then followed by

- (iv') Once these computations are done for each statistic, the estimate  $\bar{\theta}_{k+1}$  is computed by Algorithm 1 applied with  $\hat{S}_1^k$ ,  $\hat{S}_{2,j}^k$ ,  $\hat{S}_{3,j}^k$  and  $T_{k+1}$ .

*Localization procedure:* At each time step, we compute two estimators of the device position:

- (i) *Nonaveraged algorithm:* At each time step, the estimation of the device position is set as the particle with

the greatest importance weight:

$$\hat{X}_t \stackrel{\text{def}}{=} \xi_t^{i_{\max}}, \quad \text{where } i_{\max} \stackrel{\text{def}}{=} \arg\max_i \omega_t^i.$$

- (ii) *Averaged algorithm*: As the average sequence  $\{\tilde{\theta}_k\}_{k \geq 0}$  has a better rate of convergence, another particle system  $\{(\tilde{\xi}_t^i, \tilde{\omega}_t^i)\}$  is run using only step (ii) (a) above by replacing  $\theta_k$  by  $\tilde{\theta}_k$ . Then, the estimation of the device position is set as

$$\tilde{X}_t \stackrel{\text{def}}{=} \tilde{\xi}_t^{\tilde{i}_{\max}}, \quad \text{where } \tilde{i}_{\max} \stackrel{\text{def}}{=} \arg\max_i \tilde{\omega}_t^i.$$

Other natural estimators of the device position such as the posterior means could be used:

$$\hat{X}_t = \sum_{i=1}^N \omega_t^i \xi_t^i \quad \text{and} \quad \tilde{X}_t = \sum_{i=1}^N \tilde{\omega}_t^i \tilde{\xi}_t^i.$$

However, we did not observe significant differences between the estimated localization provided by the posterior means and by the proposed estimators with both simulated and true data. Moreover, we observed that the bootstrap filter sometimes produces several clouds of particles remote from each other (each of them is concentrated around local maxima of the filtering distribution). In this case, the proposed estimators offer a better accuracy than the weighted means. Finally, some indoor maps may not be convex such as in Section 5.2. In this case, the weighted means may not belong to the map while the proposed estimators always do.

*Stabilization procedure*: We add a stabilization step which consists in regularly replacing the original map estimate by the averaged one ( $\theta_k \leftarrow \tilde{\theta}_k$ ). This step is needed to improve the performance of the algorithm as detailed in Section 5.

## 5. Experiments

### 5.1. Simulated data

In this section, all experiments are performed on the finite grid  $\mathcal{C} = \{0, \dots, 30\} \times \{0, \dots, 30\}$ . Note that in the following the particles are sampled on the square  $[0, 30] \times [0, 30]$  before being associated with the corresponding cells in the finite grid  $\mathcal{C}$ . Each AP is modeled using the same coefficients  $c^*$  and  $d^*$ :

$$\forall j \in \{1, \dots, B\}, \quad c_j^* = -26 \quad \text{and} \quad d_j^* = -17.5.$$

$\Sigma$  is a Gaussian covariance function defined by  $\Sigma(x, x') \stackrel{\text{def}}{=} v_1 \exp(-|x - x'|^2 / v_2)$  with  $v_1 = 10$  and  $v_2 = 18$ . Generalities about hyper-parameter determination in spatial data modeling can be found in Cressie [10]. Ferris et al. [17] also describe the determination of the hyper-parameters  $v_1$  and  $v_2$  (when  $c_j^*$  and  $d_j^*$  are set to zero). In our case, these coefficients were calibrated after a measurement campaign in the office presented in Section 5.2. The corresponding grid stepsize is 1 m. Details about the calibration method for this particular model can be found in Dumont [15, Section 5.1]. This measurement campaign might seem contradictory with our aim to get ride of such a campaign.

However, we use this calibration to find relevant values for the true parameters  $c_j^*$ ,  $d_j^*$ . We also use this measurement campaign to estimate the hyper-parameters  $v_1$  and  $v_2$  that characterize the prior distribution of  $\delta^*$ . Their values influence the smoothness of the Gaussian field  $\delta^*$ . An online estimation of  $v_1$  and  $v_2$  could be considered but is not addressed in this paper. We assess that a partial measurement campaign on a part of the environment only could be sufficient to calibrate them. We can also consider the same values of  $v_1$  and  $v_2$  for different environments so that  $v_1 = 10$  and  $v_2 = 18$  can be used directly and no measurement campaign is needed. The variance of the observation noise is  $\sigma^{*,2} = 25$  (this value was calibrated by computing the variance of a set of RSS observations at a fixed position). The variance of the transition kernel defined in (2) is chosen such that  $a = 6$ .

All runs are started with the same initial estimates:  $\delta_0 = 0$ ,  $\sigma_0^2 = 30$  and

$$\forall j \in \{1, \dots, B\}, \quad c_{0j} = -10, \quad d_{0j} = -30.$$

The number of particles is  $N = 25$  and the initial position of each particle is chosen randomly and uniformly in  $\mathcal{C}$ . The block sizes are given by

$$\forall k \in \mathbb{N}, \quad \tau_k = 10k + 500.$$

*Mapping error*: For each map  $F_j = \mu_j + \delta_j$ , the estimation error is set as the normalized  $L_1$  error, such that the distance of a given map  $F_j$  to the true map  $F_j^* = \mu_j^* + \delta_j^*$  is

$$e_j \stackrel{\text{def}}{=} \frac{1}{|\mathcal{C}|} \sum_{x \in \mathcal{C}} |F_{j,x} - F_{j,x}^*|,$$

and the error displayed is the mean over all maps:

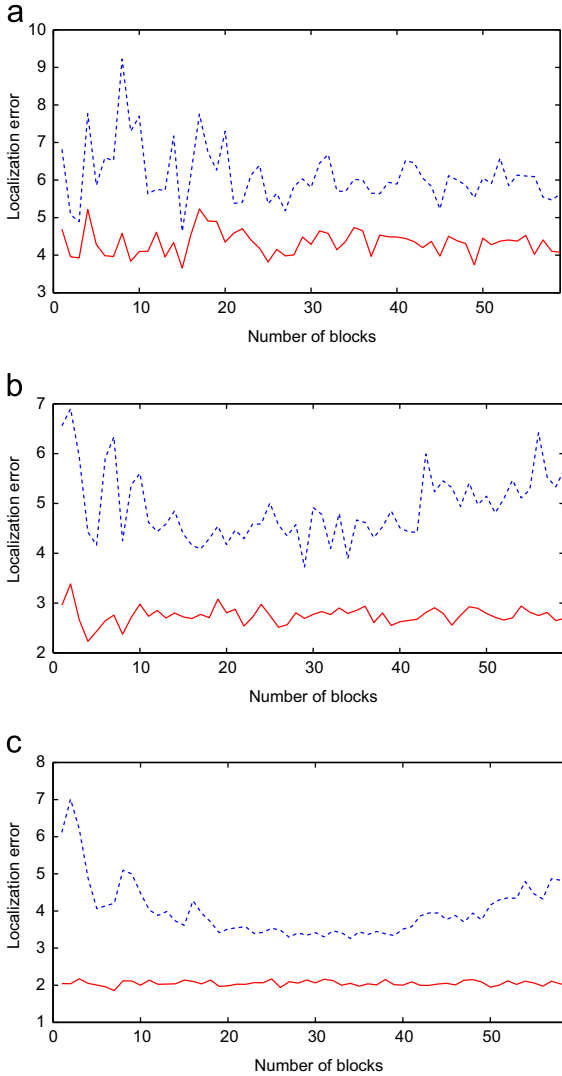
$$\bar{e} \stackrel{\text{def}}{=} \frac{1}{B} \sum_{j=1}^B e_j.$$

*Localization error*: For a given block  $\{T_k + 1, \dots, T_{k+1}\}$ , the localization error is set as the 0.8-quantile of the sample:

- $\|\hat{X}_t - X_t\|$ ,  $t \in \{T_k + 1, \dots, T_{k+1}\}$ , for the nonaveraged algorithm;
- $\|\tilde{X}_t - X_t\|$ ,  $t \in \{T_k + 1, \dots, T_{k+1}\}$ , for the averaged algorithm.

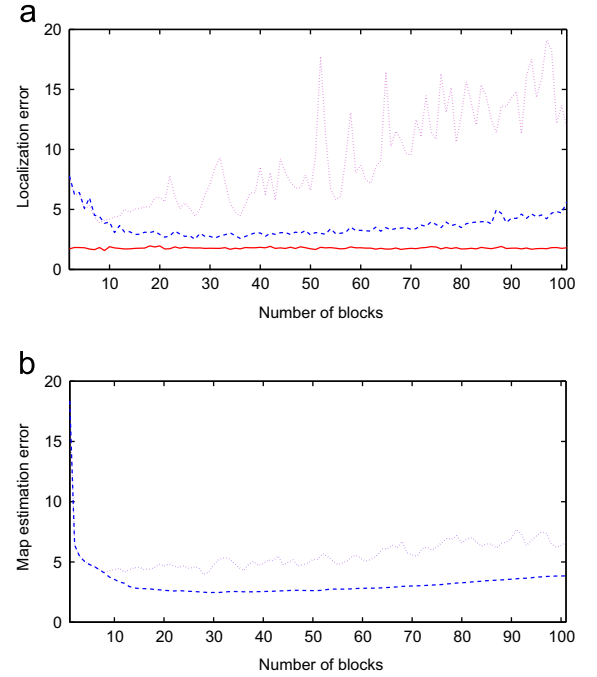
To assess the performance of our method we also display the estimated position given with a particle system run with the true maps  $F_j^*$  (i.e. using  $\theta^*$  instead of  $\theta_k$  in step ii) (a) of the algorithm). The localization error obtained using the true propagation maps  $F^*$  and the true variance  $\sigma^{*,2}$  will be referred to as the reference estimator localization error. Fig. 4 displays the localization error for a different number of AP as a function of the number of blocks when the stabilization step is omitted. As expected, both the reference estimator localization error and the localization error of the averaged algorithm are improved as the number of AP increases. However, even with a great number of AP, the localization error of the averaged estimate does not converge to the reference estimator localization error. This is confirmed by Fig. 5 which displays the map estimation error and the localization error for the greatest number of AP ( $B = 17$ ). As shown in Fig. 5a





**Fig. 4.** 0.8-quantile of the distance between the true localization and the estimated position. The localization error is given with the averaged estimate (dotted line) and the reference estimate (bold line): (a) 5 AP; (b) 10 AP; and (c) 17 AP.

the estimated position does not converge as the number of blocks (i.e. as the number of estimations) increases. After 50 blocks (about 40,000 observations) the position, which is badly estimated, does not provide good map estimates which increases the error on the averaged map estimate. Fig. 5b shows that both the map estimate and its averaged version do not converge. This convergence problem can be due to the curse of dimensionality since the number of parameters to estimate is high. Moreover, the higher the parameter space dimension is, the more likely the EM based algorithms are prone to converge towards local minima (see [7]). To overcome this difficulty, we propose to use the good behavior of the averaged estimate  $\bar{\theta}_k$  which offers a more accurate positioning than the non-averaged version  $\theta_k$  (c.f. Fig. 5). Then  $\theta_k$  is regularly replaced by the averaged version  $\bar{\theta}_k$ . In Fig. 6, this stabilization process is performed each time  $N_b = 5$  blocks



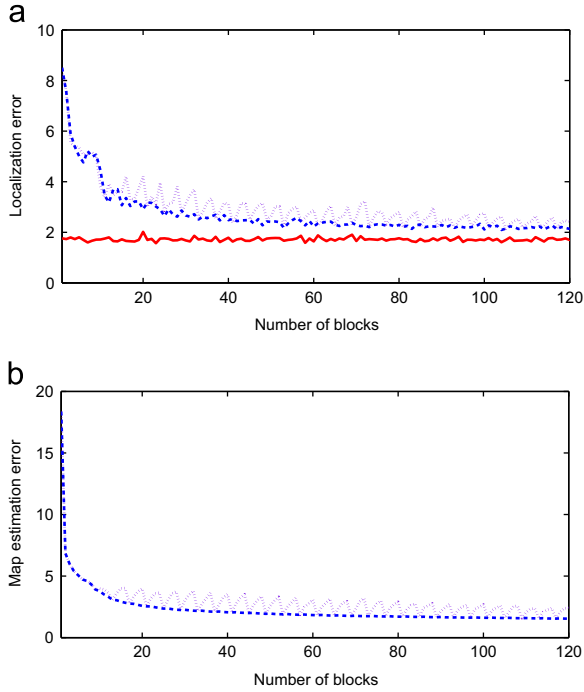
**Fig. 5.** Map estimation errors and localization errors with no stabilization step: (a) 0.8-quantile of the distance between the true localization and the estimated position. The localization error is given with the non-averaged estimate (dotted line), the averaged estimate (dashed line) and the reference estimate (bold line) and (b) mean  $L_1$  error on the map estimate with the non-averaged estimate (dotted line) and the averaged estimate (dashed line).

have been used. As shown by Fig. 6a and b, this greatly improves the performance of the estimation of the maps and of the localization. Hence, the proposed algorithm uses this stabilization procedure and the averaged estimate to localize the mobile.

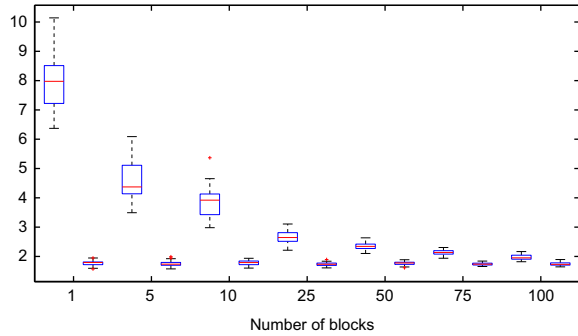
Figs. 7 and 8 illustrate the performance of the algorithm for the localization and for the estimation of the maps over 50 independent Monte Carlo runs. In Fig. 7, the reference localization error (i.e. when the maps are known) is also displayed. The convergence of the localization error to the reference error is almost reached after 100 blocks (about 100,000 observations). Similarly, the error for the estimation of the maps given by the averaged algorithm goes on decreasing after 100 blocks (the decrease is slower after 75 blocks).

## 5.2. True data

In this section, 10 AP are set up in an office environment. Fig. 9 represents a map of this environment as well as the position of the AP. The map is discretized using a grid  $\mathcal{C} \subset \{0, \dots, 32\} \times \{0, \dots, 28\}$ . Note a major difference between the model given in Section 3 and the real data situation. For any measure  $Y_k$  sent by the device, only several AP are represented in  $Y_k$ . Therefore, the maps  $\tilde{F}_j = \tilde{\mu}_j + \tilde{\delta}_j$  are not estimated simultaneously as, for any time step  $k$ , two AP might appear a different number of times in  $Y_{1:k}$ . We thus slightly modify our algorithm by introducing specific blocks and measure counters relative



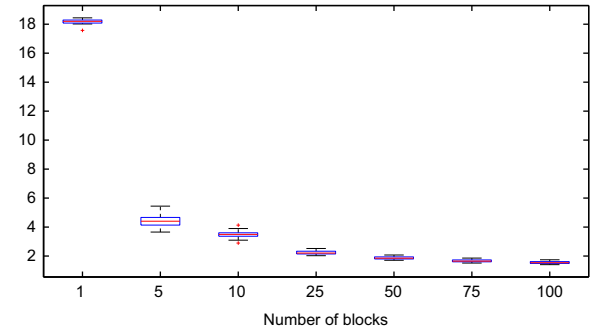
**Fig. 6.** Map estimation errors and localization errors with the stabilized algorithm: (a) 0.8-quantile of the distance between the true localization and the estimated position with the stabilization process. The localization error is given with the nonaveraged estimate (dotted line), the averaged estimate (dashed line) and the reference estimate (bold line) and (b) mean  $L_1$  error on the map estimate with the nonaveraged estimate (dotted line) and the averaged estimate (dashed line) with the stabilization process.



**Fig. 7.** Boxplots of the localization error given by the stabilized algorithm with the averaged estimate (left) and the reference estimate (right) as a function of the number of blocks.

to each AP. We shortly denote by “ $j \in Y_k$ ” the fact that AP  $j$  belongs to  $Y_k$ .

The variance  $\sigma^{*2}$  is assumed to be known and its value ( $\sigma^{*2} = 25 \text{ dBm}^2$ ) is calibrated using a measurement campaign at a fixed position. About  $T=20,000$  measures of the RSSI have been made by walking in the office for around 2 h and 45 min with a WiFi connected device (the device measures the RSSI every 0.5 s). Our algorithm produces position estimates however, unlike in the simulated data case, we do not have a direct access to the real position and thus cannot observe the localization error. To



**Fig. 8.** Boxplots of the mean  $L_1$  error on the map estimate with the stabilized algorithm and the averaged estimate as a function of the number of blocks.

overcome this difficulty, we proceed to four phases of test. During each phase, we regularly identify the true position associated with the last measure. For  $i$  in  $\{1, 2, 3, 4\}$ , we denote by  $S_{\text{test},i}$  the set of all the time steps belonging to phase  $i$ , and by  $\{X_t, Y_t\}_{t \in S_{\text{test},i}}$  the data collected during this phase of test. These data will be used to compare the estimated positions  $\{\tilde{X}_t\}_{t \in S_{\text{test},i}}$  with the true positions  $\{X_t\}_{t \in S_{\text{test},i}}$ . We will also use these data to estimate the mapping error by considering, for any  $j = \{1, \dots, 10\}$ ,

$$e_j = \frac{\sum_{i=1}^4 \sum_{t \in S_{\text{test},i}} |\tilde{F}_{j,X_t} - Y_{t,j}| 1_{j \in Y_t}}{\sum_{i=1}^4 \sum_{t \in S_{\text{test},i}} 1_{j \in Y_t}},$$

where  $1_{j \in Y_t}$  equals 1 if measure  $Y_t$  does contain AP  $j$  and equals 0 otherwise. Finally, we set

$$\bar{e} = \frac{1}{10} \sum_{j=1}^{10} e_j.$$

We run our algorithm twice on the data using the hyper-parameters  $v_1 = 10$ ,  $v_2 = 18$ . For these two experiments we will start the algorithm using different initial propagation maps:

$$F_{j,x}^0 = \mu_{j,x}^0 = c_0 + d_0 \log(\|x - O_j\|), \quad j \in \{1, \dots, 10\},$$

$c_0$  and  $d_0$  being common to all AP and  $\delta_0$  being set to zero. In the experiment 1, we consider initial parameters  $c_0 = -37$  and  $d_0 = -9$  which allow us to start the algorithm with initial estimators not too far from the real propagation maps (see Table 1). In the experiment 2, we choose  $c_0 = -37$  and  $d_0 = 9$ . In this case,  $d_0$  being positive, the initial estimators state that the further the device is from an AP, the stronger the signal will be expected. This implies that the experiment 2 starts the algorithm with a completely wrong idea about how WiFi signals propagate in the environment (see Table 1).

In the two experiments the maps  $\tilde{F}_j = \tilde{\mu}_j + \tilde{\delta}_j$  were updated a different number of times depending on the AP. This number of updates varies from two times for AP 2 to six times for AP 9. The evolution of the localization precision for the experiment 1 (resp. experiment 2) can be observed in Fig. 10 (resp. Fig. 11). Figs. 10 and 11 show that the localization improves with time for both experiments. We can observe in Fig. 10 that after a first period of improvement, the precision seems to reach a bound. On the contrary, with the experiment 2, the precision starts

really badly as we expected considering the initial point of our algorithm. The precision seems to stay constant until enough measures have been gathered by the device and until enough updates of the maps have been done. While the precision improves by around 1.5 m between the first test phase and the last one for the experiment 1, for the experiment 2, the precision considerably improves with time with a difference of 6.6 m between the beginning and the end of the experiment (see Table 2). Fig. 12 and Table 2 show that the experiment 2 final precision accuracy reaches (and even slightly overtake) the precision obtained with the experiment 1.

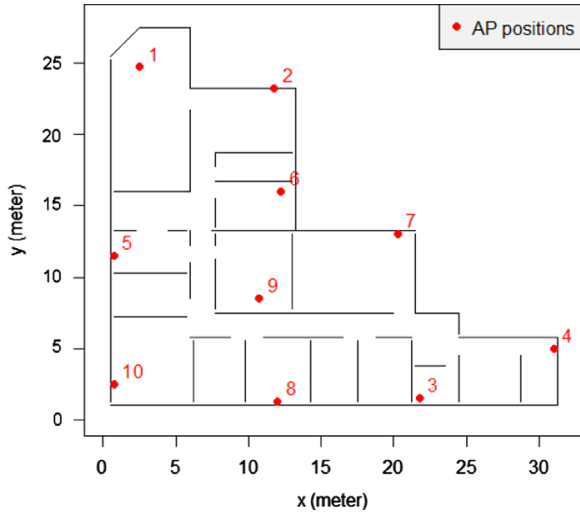


Fig. 9. Map of the indoor environment with the position of each AP (dots) and their associated identification numbers.

Table 1  
 $c_0$  and  $d_0$  parameters, initial and final mapping errors ( $\bar{e}$ ).

Experiment	$c_0$	$d_0$	Initial mapping error (dBm)	Final mapping error (dBm)
Exp. 1	-37	-9	6.4	5.1
Exp. 2	-37	9	74.5	5.2

These observations confirm the robustness of our approach. If changes occur in the environment (modifying the way WiFi signals propagate), the difference between the resulting propagation maps and the current estimator will never be as substantial as it was between the true propagation maps and the initial estimate considered in the experiment 2. We reckon that our algorithm can adapt to such changes in order to maintain the accuracy of the localization if the sufficient statistics  $\hat{S}$ ,  $k$  and  $T_k$  are regularly reset. Finally, more elaborate localization algorithms could be considered to improve the accuracy. However, the aim of this paper lies in the efficiency of our propagation maps estimation procedure for localization purpose rather than the localization algorithm itself.

## 6. Conclusion

In this paper, we propose an online EM based algorithm to estimate the propagation maps needed in any WiFi based localization system. The main difference with the existing localization solutions is that these propagation maps are estimated using the data sent by the mobile device originally used for localization purposes. The existing WiFi based localization systems establish these propagation maps either in a deterministic way or by running a previous hand-made survey. In case of environmental modifications, the propagation maps are changed. Our technique can easily adapt to these changes by regularly reinitializing the estimation procedure while hand-made survey based systems cannot take into account these modifications without renewing the survey. Other elements could be analyzed such as the size of the environment or the materials constituting the obstacles in the environment that might particularly influence the “right choice” of the hyper-parameters  $\nu_1$  and  $\nu_2$ . An online estimation procedure of these hyper-parameters could be considered. It is possible to use different localization techniques using more sophisticated human motion model for instance in order to improve positioning accuracy. However, our algorithm only needs few information on the environment, namely the size of the indoor map and the AP positions. More sophisticated models might need

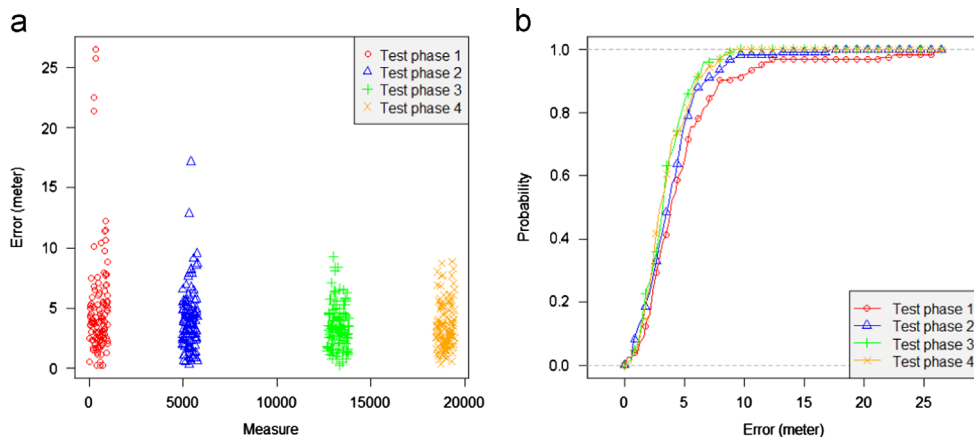


Fig. 10. Evolution of the localization precision for experiment 1: (a) tests errors and (b) tests errors CDF.

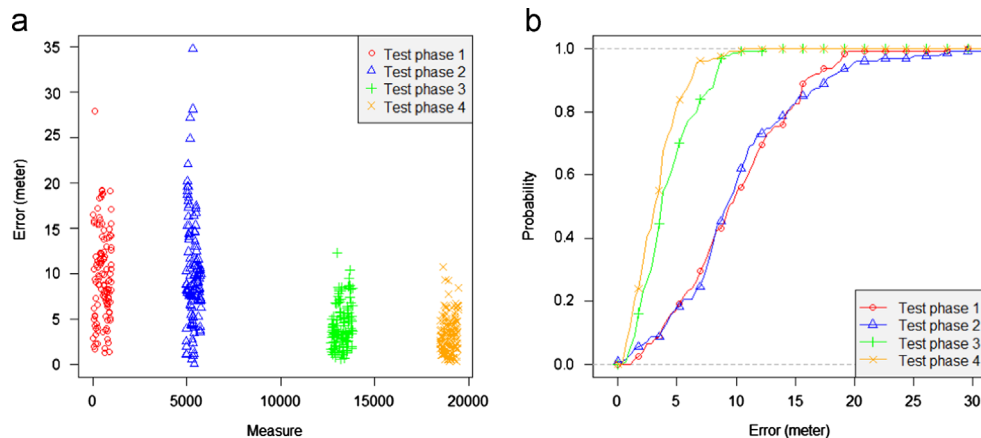


Fig. 11. Evolution of the localization precision for experiment 2: (a) tests errors and (b) tests errors CDF.

Table 2

Mean and 0.8-quantile (0.8-q) of the localization errors (in meter).

Experiment	Test phase 1		Test phase 2		Test phase 3		Test phase 4	
	Mean	0.8-q	Mean	0.8-q	Mean	0.8-q	Mean	0.8-q
Exp. 1	4.9	6.4	3.9	5.3	3.4	4.7	3.4	5.1
Exp. 2	9.9	14.3	10.1	14.4	4.2	6.5	3.3	4.9

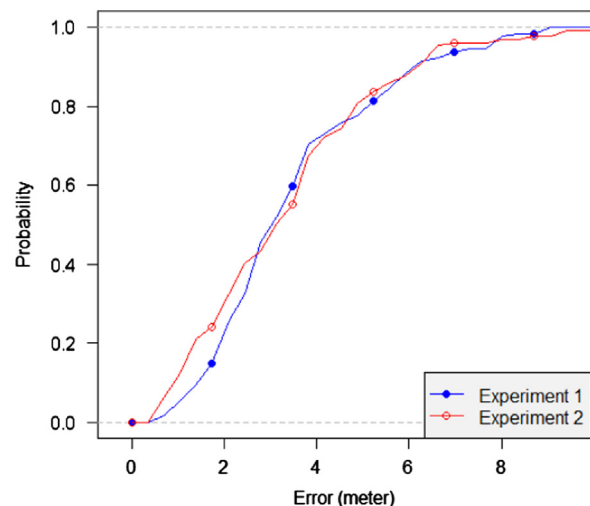


Fig. 12. CDF of the last test phase errors for the two experiments.

more information and thus make the installation process more complex.

## References

- [1] C. Andrieu, A. Doucet, S. Singh, V. Tadic, Particle methods for change detection, system identification and control, *Proc. IEEE* 92 (3) (2004) 423–438.
- [2] P. Bahl, V.N. Padmanabhan, Radar: an in-building RF-based user location and tracking system, in: 19th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, 2000, pp. 775–784.
- [3] O. Cappé, Recursive computation of smoothed functionals of hidden Markovian processes using a particle approximation, *Monte Carlo Methods Appl.* 7 (1–2) (2001) 81–92.
- [4] O. Cappé, Online sequential Monte Carlo EM algorithm, in: IEEE Workshop on Statistical Signal Processing (SSP), 2009, pp. 37–40.
- [5] O. Cappé, Online EM algorithm for hidden Markov models, *J. Comput. Graph. Stat.* 20 (3) (2011) 728–749.
- [6] O. Cappé, E. Moulines, Online expectation maximization algorithm for latent data models, *J. R. Stat. Soc. B* 71 (3) (2009) 593–613.
- [7] O. Cappé, E. Moulines, T. Rydén, *Inference in Hidden Markov Models*, Springer-Verlag New York Inc., Secaucus, New Jersey, USA, 2005.
- [8] X. Chai, Q. Yang, Reducing the calibration effort for probabilistic indoor location estimation, *IEEE Trans. Mob. Comput.* 6 (6) (2007) 649–662.
- [9] Y.-C. Chen, J.-R. Chiang, H.-h. Chu, P. Huang, A.W. Tsui, Sensor-assisted Wi-Fi indoor location system for adapting to environmental dynamics, in: Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, ACM, New York, 2005, pp. 118–125.
- [10] N. Cressie, *Statistics for Spatial Data*, Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, John Wiley, Hoboken, New Jersey, USA, 1993.
- [11] P. Del Moral, *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer, New York, 2004.
- [12] P. Del Moral, A. Doucet, S. Singh, Forward Smoothing Using Sequential Monte Carlo, Technical Report, 2010, arXiv:1012.5390v1.
- [13] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. R. Stat. Soc. B* 39 (1) (1977) 1–38. (with discussion).
- [14] A. Doucet, A. Johansen, A tutorial on particle filtering and smoothing: fifteen years later, in: *Oxford Handbook of Nonlinear Filtering*, 2009.
- [15] T. Dumont, Contributions à la localisation intra-muros, de la modélisation à la calibration théorique et pratique d'estimateurs (Ph.D. thesis), Université Paris-Sud, Orsay, 2012.
- [16] F. Evennou, F. Marx, Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning, *EURASIP J. Appl. Signal Process.* 2006 (2006) 1–12.
- [17] B. Ferris, D. Hähnel, D. Fox, Gaussian processes for signal strength-based location estimation, in: *Robotics: Science and Systems*, The MIT Press, Cambridge, Massachusetts, USA, 2006.
- [18] H.T. Friis, A note on a simple transmission formula, *Proc. IRE* 34 (5) (1946) 254–256.
- [19] J.-M. Gorce, K. Jaffres-Runser, G. de la Roche, Deterministic approach for fast simulations of indoor radio wave propagation, *IEEE Trans. Antennas Propag.* 55 (March (3)) (2007) 938–948.
- [20] N. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear and non-Gaussian Bayesian state estimation, *IEE Proc. Radar Signal Process.* 140 (2) (1993) 107–113.
- [21] S. Le Corff, G. Fort, Convergence of a particle-based approximation of the block online expectation maximization algorithm, *ACM Trans. Model. Comput. Simul.* 23 (1) (2013) 21–222.
- [22] S. Le Corff, G. Fort, Online expectation maximization based algorithms for inference in hidden Markov models, *Electron. J. Stat.* 7 (2013) 763–792.

- [23] S. Le Corff, G. Fort, E. Moulines, Online EM algorithm to solve the SLAM problem, in: *IEEE Workshop on Statistical Signal Processing (SSP)*, 2011, pp. 225–228.
- [24] D. Madigan, E. Einahrawy, R.P. Martin, W.-H. Ju, P. Krishnan, A. Krishnakumar, Bayesian indoor positioning systems, in: *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, IEEE, 2005, pp. 1217–1227.
- [25] G. Mongillo, S. Denève, Online learning with hidden Markov models, *Neural Comput.* 20 (7) (2008) 1706–1716.
- [26] S. Seidel, T. Rappaport, 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings, *IEEE Trans. Antennas Propag.* 40 (2) (1992) 207–217.